2012 International Conference on Communication, Information & Computing Technology (ICCICT), Oct. 19-20, Mumbai, India

Priority-based Congestion Control Protocol (PCCP) for Controlling Upstream Congestion in Wireless Sensor Network

Dipti Patil
Computer Engineering Department
MGM's College of Engineering and Technology
Navi Mumbai, INDIA
diptipatil@mgmmumbai.co.in

Sudhir N. Dhage
Computer Engineering Department
Sardar Patel Institute of Technology
Andheri(West), Mumbai, INDIA
Sudhir_dhage@spit.ac.in

Abstract— **Congestion in Wireless Sensor Network (WSN) occurs when traffic load allocated to any sensor node is beyond its capacity. It is necessary to control the congestion traffic to support traditional quality of service (such as packet loss ratios, packet delay, wasting energy and throughput) especially, multimedia application of Wireless Multimedia Sensor Network (WMSN). Congestion control in WSN has been shown to be very effective with respect to extending lifespan of system. Hence to avoid the congestion in WSN, we proposed an efficient protocol called Priority-based Congestion Control Protocol (PCCP) which prevents an upstream congestion in WSN. The PCCP creates priority table based an importance of each node, and then sends this information to all the nodes within the network. Secondly, the major function of PCCP is to measure congestion level in the network using ratio of packet inter-arrival time along over packet service time. PCCP uses to control upstream congestion along with congestion degree and packet table. PCCP is hop-by-hop upstream congestion control protocol for WSN. PCCP provides work for packet-based computation to optimize congestion control. It can work under both single path & multi-path routing. PCCP ensures guarantee of packet loss as well as delay, resulting each node can avoid unfairness and achieve flexible throughput. In this paper, we have proposed that PCCP provides better QoS by controlling network recourse management and network traffic to consuming efficient energy.**

Keywords-Congestion contro; Wireless Sensor Network (WSN); Upstream Congestion; Congestion Traffic; Quality Of Services.

I. INTRODUCTION

This Sensor Network generally does not have pre–determine topology. Indeed of this, the sensor nodes construct & dynamically maintain the structure of the network through wireless communication. These nodes transmit and deployed gathered data over wide areas to one or several central nodes, which is also called the base station/sink. A WSN consist of one or more sinks and perhaps ten or thousand of sensor nodes scattered in an area.

Wireless sensors network (WSN) consist many sensor nodes to co-operatively reporting environmental situations within a region where the environment abruptly changes, such as temperature, sound, vibration, pressure, motion or pollution. These nodes act like a routers and simultaneously collects the information/data to transmit [1]. Under light load the data traffic in the network is light. But if the load becomes heavy and the data traffic also increase beyond the capacity. This might leads to congestion.

Congestion can occurs in WSN because of many sources like buffer overflow, concurrent transmission, packet collision and many to one nature. Congestion generate in network because of access load is allocated on a link /node called as traffic, Fig. 1, There are two type of traffic: i) Downstream traffic: The downstream traffic from sink to the sensor nodes usually is a one-to many multi-casts. ii) Upstream traffic: Upstream traffic from sensor nodes to sink is a many to one multi-hop convergent. Therefore congestion most probability appears in the upstream direction [2].

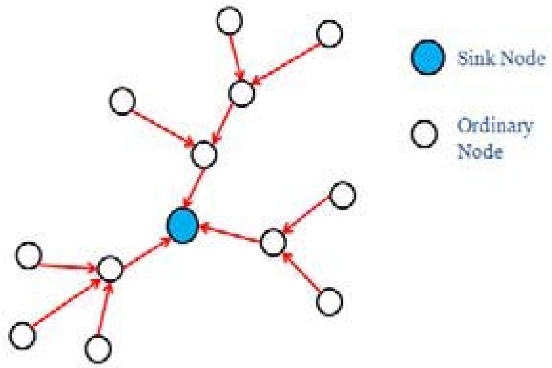

Figure 1. Many to one traffic pattern.

The upstream traffic can be classified into 4 categories:

- Event-based - A sensor node reporting only if target events occur.
- Continuous - Sensor nodes may need to periodically report to the sink and generate continuous data transmission in some cases.
- Query-based - In query-based delivery, sensory data is stored inside network and transmitted to the sink on demand.

- Hybrid - Practical applications might trigger hybrid data delivery including event-based, continuous, and query-based [2].

The rate of packet drops at the base node is directly proportional to source rate i.e. when the source rate increase beyond a certain threshold capacity, congestion occurs which causes packet drops and ultimately increasing delay. Therefore it degrades Quality of Services. Hence we need to control such congestion to get flexible throughput and better QoS.

There are two types of congestion could occur in WSN as shown in Fig. 2. They are:

- Node-level congestion: It is caused by buffer overflow in the node and can result in packet loss, and increased queuing delay.
- Link-level congestion increases packet service time, and decreases both link utilization and overall throughput, and wastes energy at the sensor nodes.

Both node-level and link-level congestions have direct impact on energy [3].

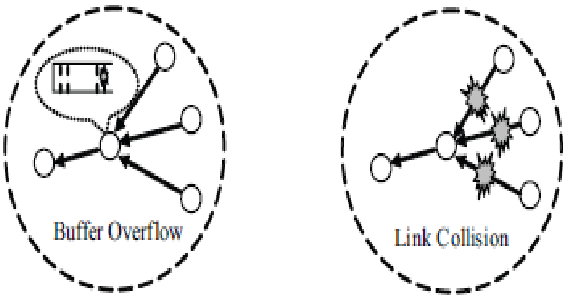

Figure 2. Node level congestion Link Level congestion

Congestion control generally follows three steps: a)any detection b) congestion notification c) rate adjusting. Therefore we need to find protocol which detect & control congestion whose efficiency depends on how much it can achieve energy efficiency and how much it can support to the quality of services (QoS) [3].

Two general approaches to control congestion they are network resource management and traffic control. Network resource management tries to increase network resource to mitigate congestion when it occurs. In WSN, multiple radio interfaces can be used to increase bandwidth and weaken congestion. It is necessary to guarantee precise and exact network resource adjustment in order to avoid over- provided resource or under-provided resource[4][5]. In this paper we focus on upstream traffic in WSNs, traffic control implies to control congestion through adjusting traffic rate at source node or intermediate nodes. According to the control behavior of upstream traffic there are two general method of traffic control in WSNs. i) End-to-end: the end-to-end control can impose exact rate adjustment at each source node and simplify the design at intermediate nodes. It give slow response and relies highly on round-trip time (RTT). ii)Hop-by-hope: The hop-by-hope congestion control has faster response but difficult to adjust the packet-forwarding rate[6].

In this paper we have propose a priority based congestion control protocol (PCCP), which employs packet-based computation to optimized congestion control in WSN. The PCCP creates priority table based on importance of each node and sends this information to all the necessary nodes within the network [7]. To measures congestion level it refers congestion degree which is the ratio of service time over inter arrival time.

The main objective is to examine the behaviour of several network parameters & their impact to congestion which is to be control and avoid. The simulation scenarios were implemented and tested with the use of the NS-2 simulator.

The reminder of this paper is organized as follows:

Section II introduces related work. Section III propose objective of this paper & define problem statement. Section IV represents an overview of overall protocol design. Section V describes simulation set-up and simulation result. Section VI Concludes the paper and introduced for future work.

II. RELATED WORK

Our aim to select best technique which provides high throughput with low possible delays and increases energy efficiency. Hence, we try to minimize the delay for route which will help us to reduce the round trip time (RTT) for a given network topology for this we use ad- hoc network which are important considerations due to the unpredictability and heterogeneity of link qualities[8][9].

In recent years, many new link quality metrics have been proposed. The first technique measures the round trip delay between neighbouring nodes and proposes Per-hop Round Trip Time (RTT). Per-hop Packet Pair Delay (PktPair) measures the delay between a pair of back-to-back probes to a neighbour node [10]. Expected Transmission Count (ETX) measures the loss rate of broadcast packets between pairs of neighbouring nodes and estimates the number of retransmissions required to send unicast packets [11]. Weighted Cumulative Expected Transmission Time (WCETT) is used for selecting channel-diverse paths and accounts for the loss rate and bandwidth of individual links [12]. Expected Data Rate (EDR), for accurately finding high-throughput paths in multi-hop ad hoc wireless networks for transmission interference. Unfortunately, none of these metrics can be directly applied to wireless sensor network that simultaneously take into account delay, throughput and interference.

Adaptive rate control (ARC), is an LIMD-like (linear increase and multiplicative decrease) algorithm [9]. In ARC, if the intermediate node send the information to its parent node that the previous packets are forwarded successfully, then the parent node increasing its packet transfer rate with constant amount α. Or it increases its rate by multiplying the constant factor β with initial rate where the β is always lies between $0 < \beta < 1$. ARC does not use explicit congestion detection or explicit congestion notification[5]. In an event-to-sink reliable transport protocol (ESRT) is proposed for congestion control [3]. ESRT is a centralized protocol that regulates the reporting rate of sensors in response to congestion detected by a sink. When the sink receives a packet with the congestion notification bit set, it infers congestion and broadcasts a control

978-1-4577-2078-9/12/$26.00©2011 IEEE

signal notifying all source nodes to reduce their reporting frequency. In fusion, according to queue length congestion is detected in each sensor node. The node at which congestion detects, consider as congestion notification (CN) bit. When CN bit is set, neighbouring nodes get alert and stop forwarding packets to such congested node and drain the backlogged packets. It is inefficient in link utilization and fairness [4].

Congestion control & fairness (CCF) uses packet service time to deduce the available service rate and therefore detects congestion in each intermediate sensor node [7]. CCF controls congestion in a hop-by-hop manner and each node uses exact rate adjustment based on its available service rate and child node number. In CCF each node receives the same throughput therefore it gives guarantees simple fairness. But the CCF relies only on packet service time which could lead to low utilization when some sensor nodes do not have enough traffic or there is a significant packet error rate (PER). Congestion detection & avoidance (CODA) [8] detects congestion based on buffer capacity and wireless channel load. CODA designs both open loop and closed-loop rate adjustment, and the algorithm used to adjust traffic rate works in a way like additive increase multiplicative decrease (AIMD)[4].

In Siphon - Overload Traffic Management using Multi-radio Virtual Sinks in Sensor Networks proposes to exploit the availability of a small number of all wireless, multi-radio virtual sinks that can be randomly distributed or selectively placed across the sensor field [4]. The siphon also infers congestion based on queue length in intermediate nodes [9]. But it uses traffic re-direction to weaken congestion. It supports a variety of application specific line aggression, low delay transport strange and localized activation. But there is no rate adjustment in siphon.

In WSN, all above congestion control protocol are provides only guarantee of simple fairness & supports only single path routing. So, we need a protocol which provides high throughput with lowest delays, improve energy efficiency & reliability and supports single path as well as multipath routing.

III. PROPOSE SYSTEM

This section describes MAC protocol, network model and node model for controlling upstream congestion as shown in following figures.

MAC layer reliability for congestion control: Generally MAC protocol is considered as auto-rate protocol. They use queue length to get estimate of congestion in the network. Assumes that the nodes total buffer size is Q and the current number of packets in the buffer is q. When the buffer is full, i.e. Q = q, the node is congested completely, the packets arrived at this node will be discarded. Conversely, when the buffer is not full, the input packet rate R_{in} and output packet rate R_{out} are monitored.

R_{in} is the reciprocal of ΔT_{in}, i.e.

$$R_{in} = 1/\Delta T_{in} \qquad (1)$$

Where ΔT_{in} represents the packet arrival interval.

Packet arrival interval s referred as the time interval of two consecutive packets received at the node MAC layer. Rout is the reciprocal of ΔT_{out} ,i.e.

$$R_{out} = 1/\Delta T_{out} \qquad (2)$$

Where, ΔT_{out} represents the packet service time.

Packet service time is referred as the time interval between the time that a packet arrives at MAC layer and the time that it is transmitted successfully, ΔT_{out} is the sum of the time for queue, collision, back off and transmission. This protocol acts like the receiver, measures its congestion condition and sends the congestion information back to the sender along with the transmission rate based on the channel condition and the sender transmits a limited number of back-to-back data packets at the selected rate according to the feedback information.

A. *Network model :*

This paper addresses upstream congestion control for a WSN that supports single-path and multi-path routing. Figure represents the network model where sensor nodes are supposed to generate continuous data and form many-to-one convergent traffic in the upstream direction. The MAC layer implemented CSMA/CA MAC protocol which receives congestion condition and send this information to the sender. Each sensor node could have two types of traffic: i) Source traffic: The traffic which is locally generated at each sensor node is called sours traffic. ii) Transit traffic: The traffic generated by other nodes (neighboring nodes) . As shown in Fig.3 node 1 is a source node so that it has only one type of traffic called source traffic, while nodes 2,3,4,5,6 and 7 are source nodes as well as intermediate nodes Therefore they have both traffics ,source traffic as well as transit traffic. Each node could have two types of neighbor nodes: backward and forward. For example, the backward node of node 2 is nodes 1 while, foreword node of node 2 are node 4 and node 5. Forward nodes are denoted by letter f and represented as f(*i*) where *i* is node number. Backward nodes are denoted by letter b and represented as b(*i*). For node 2 has forward nodes are {4,5} and backward nodes are {1}.

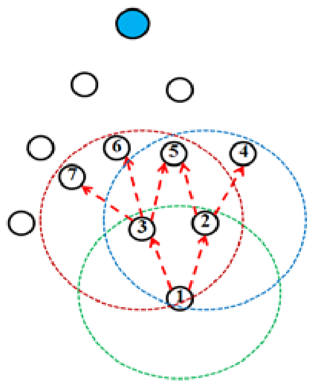

Figure 3. Network Model

B. *Node model :*

In Fig. 4, *i* be the node in WSN. In MAC layer the transit traffic of node *i* is r^{i}_{tr} which is received from its child nodes such as node *i*−1. Before forwarding the packets from node I to its next node *i*+1 both the transit traffic and the source traffic converge at the network layer by the parent node of *i*. The total input traffic rate of node *i* at MAC layer is,

$$r^i_m = r^i_{src} + r^i_{tr} \tag{3}$$

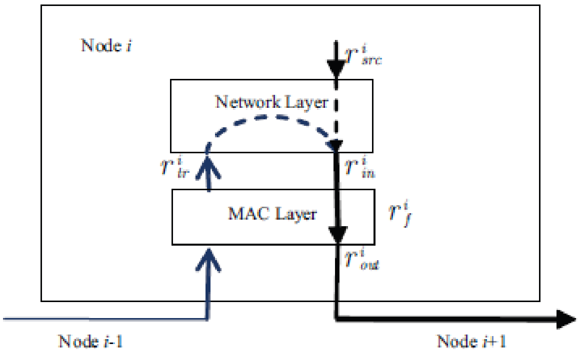

Figure 4. The queuing model at a particular sensor node

When this total input traffic rate r^i_m greater than packet forwarding rate r^i_f packets could be queued at MAC layer. The packet output rate at the node *i* is r^i_{out} which is forwards to its next node *i*+1. If r^i_{in} is smaller than r^i_f, then r^i_{out} is equals to r^i_{in} i.e.

$$\text{If} \quad r^i_{in} < r^i_f$$
$$\text{then,} \quad r^i_{out} = r^i_{in} \tag{4}$$

Otherwise, If r^i_{in} is greater than r^i_f, then r_{out} will be close to r^i_f i.e.

$$\text{If} \quad r^i_{in} > r^i_f$$
$$\text{then,} \quad r^i_{out} != r^i_{in} \tag{5}$$
$$\text{But,} \quad r^i_{out} \text{ close to } r^i_{in}$$

Therefore, we can say,

$$r^i_{out} = \min(r^i_{in}, r^i_f) \tag{6}$$

Indirectly we reduce r^i_{out} by reducing r^i_{in} we adjust r^i_f and r^i_{in} en such a way that r^i_{in} is indirectly proportional to r^i_f. We can reduce r^i_{in} and increase r^i_f.

Through MAC protocol we can increase r^i_f, therefore it is easier to low r^i_{in}. We can change the source rate by changing sampling frequency. In fact, the output traffic at node *i* is part of transit traffic at the node *i*+1. Therefore reduction of r^i out implies a decrease of r^i+1 . If packet input rate r^i in greater than packet forwarding rate r^i_f, then there will be backlogged packets inside node *i* and node-level congestion takes place. So that, we need to reduce r^i in and/or increase r^i_f. While r^i_f can be increased through adjusting MAC protocols, it is much easier to lower r^i in through throttling either r^i_s, r^i_{tr} or both of them. The source rate r^i_s can be reduced locally by changing sampling (or reporting) frequency. The transit traffic r^i_{tr} can be indirectly reduced through rate adjustment at the node $i-1$. On the other hand, if there is collision on the link around the node *i*, then node *i* and its neighbouring nodes should reduce channel access in order to prevent further link-level congestion. Although this task may be performed through the MAC, yet it is easier to reduce r^i_{in}.

IV. Protocol Description

In WSNs, congestion degrades quality of channels, loss of packets and consumes excess energy which leads to buffer drops and ultimately increasing delay. In WSNs, sensor nodes might have different priority due to their function or location. Therefore congestion control protocols need guarantee weighted fairness so that the sink can get different, but in a weighted fair way, throughput from sensor nodes. With the fact that multi-path routing is used to improve system performance of WSNs, congestion control protocols need to be able to support both single-path routing and multi-path routing. Congestion control protocols need to support traditional QoS in terms of packet delivery latency, throughput and packet loss ratio, which is required by multimedia application in WMSNs and in tern improves the energy efficiency. It tries to guarantee weighted fairness and supporting multi-path routing with lower control overhead. In this section, we discuss the detail of our schema including A) Congestion detection B) congestion feedback / notification C) Priority-based rate adjustment.

A. Congestion Detection (CD)

In congestion detection (CD) detect congestion detects congestion based on mean packet inter-arrival (t^i_a) and mean packet service times (t^i_a) at the MAC layer. Here packet inter-arrival time is defined as the time interval between two sequential arriving packets from either source or for the transit traffic, and the packet service time is referred to as the time interval between when a packet arrives at the MAC layer and when its last bit is successfully transmitted. t^i_s covers packet waiting, collision resolution, and packet transmission times at the MAC layer. t^i_a a as well as t^i_s can be measured at each node i on a packet-by-packet basis.

Based on the t^i_a and t^i_s, ICD defines a new congestion index, congestion degree d (*i*), which is defined as the ratio of average packet service time over average packet inter-arrival time over a pre-specified time interval in each senor node i as follows:

$$D(i) = t^i_a / t^i_s \tag{7}$$

The congestion degree is intended to reflect the current congestion level at each sensor node. When the inter-arrival time is smaller than the service time, the congestion degree d(*i*) is larger than 1 and the node experiences congestion. Otherwise when the congestion degree d(*i*) is smaller than 1, the incoming rate is below the outgoing rate, and hence congestion abates. Therefore congestion degree can adequately represent congestion condition and provide helpful information in order to realize efficient congestion control. The congestion degree d(*i*) can inform the child nodes about the traffic level to be increased or decreased by adjusting their transmission rate. In Eq. (1), t^i_a and t^i_s at each node *i* are measured using EWMA (exponential weighted moving average) algorithm as follows.

In the process of determining the congestion degree, t^i_a is updated periodically whenever there are *Np* (=100 in PCCP) new packets arriving as follows:

$$t^i_a = (1 - w_a) * t^i_a + W_a * T_{Np} / N_p \tag{8}$$

where $0 < w_a < 1$ is a constant (= 0.1 in PCCP examples to be discussed later), *TNp* is the time interval over which the measurements are performed, and within which the *Np* new packets arrive.

Also, $t^i{}_s$ is updated each time a packet is forwarded as follows:

$$t_s = (1 - w_s) * t_s + w_s *t^i{}_s \qquad (9)$$

where 0 < ws < 1 is a constant (again set to 0.1 in the future examples), ti s is the service time of the packet just transmitted.

B. Cogestion Feedback

This protocol uses implicit congestion notification. Each node *i* piggybacks the packet scheduling rate; the number of child nodes, and packet service rate, in its packet header. All the child nodes of node *i* overhear the congestion notification information. Whenever the value of packet service ratio of parent *j* of node *i* denoted by (lowers the threshold or greater than 1, multipath rate control procedure (explained in the next section) is triggered.

C. Priority-based Rate Adjustment

Our multipath rate control scheme uses hop by hop rate adjustment for multiple paths. In this protocol, the output rate of a node is controlled by adjusting the scheduling rate, r_{sch} .In fact, by adjusting the scheduling rate for parent *j*, $r^i{}_{jsch}$; the packet loss due to buffer overflow is avoided and automatically the total scheduling rate, $r^i{}_{sch}$ is adjusted as scheduling rate is the sum of scheduling rate for all the parents *j* of node *i*.

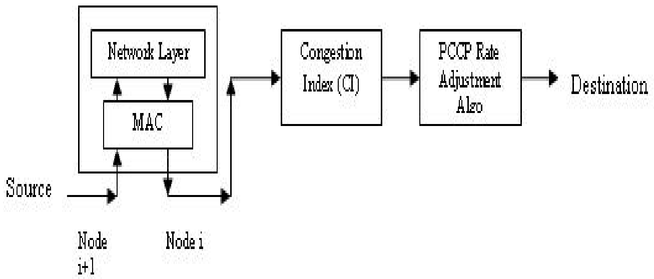

Figure 5. Architectural Model of PCCP

The algorithm works as follows:

1) Initially, the packet service ratio for each node *i* will be 1, each node will have a lower scheduling rate, $r^i{}_{sch}$; and the scheduling rate for each of its parents will be distributed i.e. it find packet arrival time, packet service time and packet service rate $r^i{}_{scr}$.

2) Each node *i* will adjusts its scheduling rate towards multiple parents by calling procedure. At first each node *i* calculates its packet service ratio as well as its parents i.e. in this phase priority index is determine and check.

3) Calculating the scheduling rate, each node, *i* update their originating rate according to the method. The originating rate depends on the scheduling rate as well as on the priority for each application data requested by the sink.

V. SIMULATION ENVIRONMENT

We have performed extensive simulations to evaluate the performance of our scheme. We determine the threshold value of packet service ratio, μ with the simulation.The version in use for this dissertation is version 2.32, with installation of the all-in-one package that operates on Linux. The routing protocol is implemented using C++ and the scenarios are implemented with scripts written in TCL that comprise commands and parameters for simulator initialization, node creation and configuration. The basic parameters needed for simulation are the movement pattern file, the communication pattern file and the configuration file. The movement pattern file describes all node movements while the communication pattern file describes the packet workload presented at the network layer during simulation. These two files essentially constitute the description of the simulation scenario. The final input is the configuration file that defines the ad hoc network routing protocol which is often the main file where the scenario files are called. The procedure for running the scenarios is shown in Fig.6:

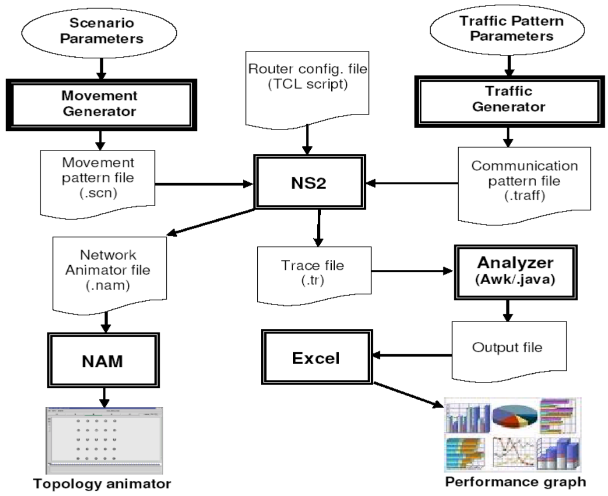

Figure 6. Flow Diagram for Running Scenario in NS-2

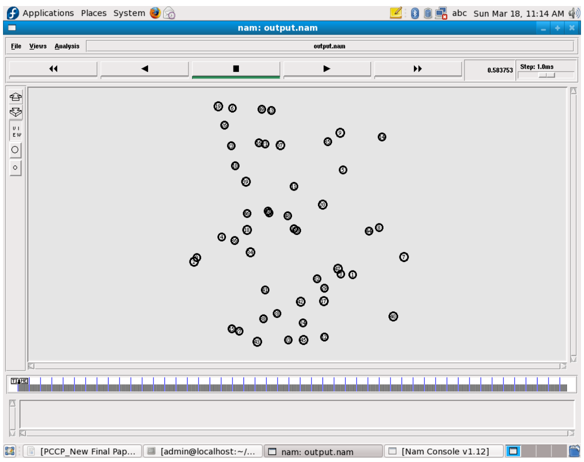

Figure 7. Running Scenario in NS-2: Stage 1

As described above there are two types of simulation results a text based output file known as trace file and graphical based file. The major component of the Ns-2 simulator is the event scheduler. Each packet in Ns-2 is unique and has its own

Packet ID. The event scheduler recognizes packet by its Packet ID and fire all the events in the event scheduler queue for the current time invoking the appropriate network components.

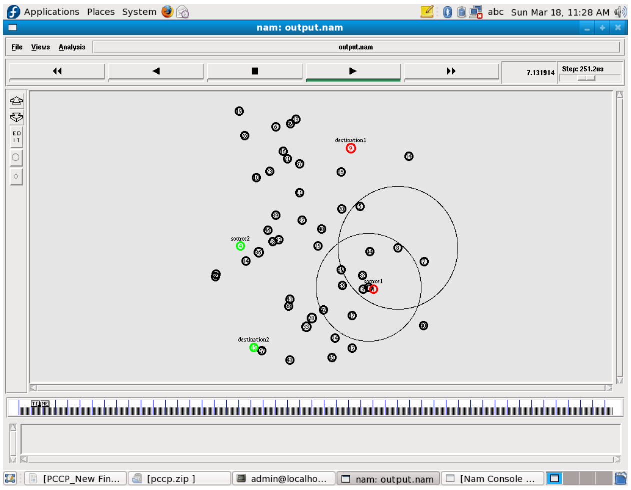

Figure 8. Running Scenario in NS-2: Stage 2

The text based output contained the details of simulation data and can be analysed by programming code to get the required information. The trace file can be used to record individual packet information as it arrives, departs or dropped at the link or queue. The NS-2 contain various type of trace format suitable for different type of simulation such as wireless trace format (version 1 and 2), AODV routing algorithm trace format , DSR routing algorithm trace format and others. The second output file is basically an input file to a graphical simulation tool known as Network Animator (NAM) which assists the user to get more information about their simulation by visualizing packet trace data. NAM can graphically display information such as throughput and number of packet drops at each link, although the graphical information cannot be used for accurate analysis. The results from the simulation are generated in files, an output trace file (*.tr). The trace file will contain information on the various events which occurred, details of node behavior, packet transmissions and receptions etc. Analysis of these packets can determine the performance effects from parameter variation, routing protocols and are performed using awk scripts.

VI. CONCLUSIONS

In this paper, we are proposing a distributed and scalable technique that eliminates congestion within wireless sensor network, and that will ensure the fair delivery of packets to the destination node, or base station. We say that fairness is achieved when equal numbers of packets are received from each node. Since in general, we have many sensors transmitting data to the base station.

So, proposed hop-by-hop upstream congestion control protocol for WSN, called PCCP. Congestion control protocols will avoid or reduce packet loss due to buffer overflow, and remain lower control overhead that consumes less energy. It's support traditional QoS metrics such as packet loss ratio, packet delay, and throughput. It will propose fairness to guarantee so that each node can achieve fair throughput. Most of the existing work [3] [4] guarantees simple fairness in that every sensor node obtains the same throughput to the sink. In fact, sensor nodes might be either outfitted with different sensors or geographically deployed in different place and therefore they may have different importance or priority and need to gain different throughput. PCCP will detect congestion jointly using packet inter-arrival and service times. The node priority index and realizes weighted fairness; it works for both single-path and multi-path routing. PCCP will achieve high link utilization and flexible fairness. PCCP will lead to small buffer size; it will avoid/reduce packet loss and in turn improves energy-efficiency, and will provide lower delay. Compare the Priority-based congestion control protocol (PCCP) with congestion control fairness (CCF) performance and calculate to detect congestion on packet inter - arrival time.

REFERENCES

[1] Rekha Chakravarthi1, C. Gomathy2, Suraj K. Sebastian3, Pushparaj. K4, Vinu Binto Mon5 "A Survey on Congestion Control in Wireless Sensor Networks", International Journal of Computer Science & CommunicationVol. 1, No. 1, January-June 2010, pp. 161-164.

[2] Saeed Rasouli Heikalabad1, Ali Ghaffari2, Mir Abolgasem Hadian2 and Hossein Rasouli; "DPCC: Dynamic Predictive Congestion Control in Wireless Sensor Networks'' IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 1, January 2011.

[3] R.Then Malar; "Congestion Control in Wireless Sensor Networks Based Multi-Path Routing In Priority Rate Adjustment Technique", International Journal of Advanced Engineering & Applications, Jan. 2010.

[4] R.Gunasundari, Department of Electronics and Communication Engineering, Pondicherry Engineering College, Puducherry, "An Efficient Congestion Avoidance Scheme for Mobile Healthcare Wireless Sensor Networks", Int. J. Advanced Networking and Applications Volume: 02, Issue: 03, Pages: 693-698 (2010).

[5] C. Wang, Member, IEEE, B. Li, Senior Member, IEEE, K. Sohraby, Senior Member, IEEE, M. Daneshmand, Member, IEEE, and Y. "Upstream Congestion Control In Wireless Sensor Networks Through Cross-Layer Optimization" Ieee Jaournal On Selected Areas In Communications, Vol. 25, No. 4, May 2007.

[6] Douglas S. J. De Couto, Daniel Aguayo, Benjamin A. Chambers, and Robert Morris, "Performance of multihop wireless networks: shortest path is not enough", SIGCOMMComputer Communication Rev., 33(1):83-88, 2003.

[7] Chonggang Wang1, Kazem Sohraby1, Victor Lawrence2, Bo Li3, Yueming Hu, "Priority-based Congestion Control in Wireless Sensor Networks"The IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06) 0-7695-2553-9/06.

[8] J. Park and S. Kasera. "Expected data rate: an accurate high-throughput path metric for multi-hop wireless routing", In Proc. of second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2005.

[9] Jang-Ping Sheu 1, 2, Li-Jen Chang2 and Wei-Kai Hu2, "Hybrid Congestion Control Protocol in Wireless Sensor Networks", Journal Of Information Science and Engineering 25, 1103-1119 (2009).

[10] Muhammad Mostafa Monowar, Md. Obaidur Rahman and Choong Seon Hong, "Multipath Congestion Control for Heterogeneous Traffic in Wireless Sensor Network", Department of Computer Engineering, Kyung Hee University, South Korea, Feb. 17-20, 2008 ICACT 2008.

[11] C.T. Ee and R. Bajcsy, "Congestion control and fairness for many-to-one routing in sensor networks," in Proc. ACM Sensys, pp.461-473, Nov. 2004.

[12] Holger, Andreas Willing, "Protocols and architectures for wireless sensor networks".